```
#include <iostream>
using namespace std;

// Defining region codes
const int INSIDE = 0; // 0000
const int LEFT = 1; // 0001
const int RIGHT = 2; // 0010
const int BOTTOM = 4; // 0100
const int TOP = 8; // 1000

// Defining x_max, y_max and x_min, y_
// clipping rectangle. Since diagonal
// enough to define a rectangle
const int x_max = 10;
const int y_max = 8;
const int x_min = 4;
const int y_min = 4;

// Function to compute region code for
int computeCode(double x, double y)
{
    // initialized as being inside
    int code = INSIDE;

    if (x < x_min) // to the left of r
        code |= LEFT;
    else if (x > x_max) // to the righ
        code |= RIGHT;
    if (y < y_min) // below the rectar
```

```
#include<graphics.h>
#include<conio.h>
#include<stdio.h>
#include<math.h>
void main()
{
int rcode_begin[4]=
{0,0,0,0},rcode_end[4]=
{0,0,0,0},region_code[4];
int
W_xmax,W_ymax,W_xmin,W_ymin,flag=0;
float slope;
int x,y,x1,y1,i, xc,yc;
int gr=DETECT,gm;
initgraph(&gr,&gm,"C:\\TURBOC3\\BGI");
printf("\n****** Cohen Sutherlsnd Line
Clipping algorithm ***********");
printf("\n Now, enter XMin, YMin =");

 scanf("%d %d",&W_xmin,&W_ymin);
scanf("%d %d",&W_xmin,&W_ymin);
printf("\n First enter XMax, YMax =");
scanf("%d %d",&W_xmax,&W_ymax);
printf("\n Please enter intial point x
and y= ");
scanf("%d %d",&x,&y);
printf("\n Now, enter final point x1
and y1= ");
scanf("%d %d",&x1,&y1);
cleardevice();
rectangle(W_xmin,W_ymin,W_xmax,W_ymax)
;
line(x,y,x1,y1);
line(0,0,600,0);
line(0,0,0,600);
if(y>W_ymax)  {
rcode_begin[0]=1;     // Top
flag=1 ;
}
if(y<W_ymin) {
rcode_begin[1]=1;            // Bott
```

```
flag=1;
}
if(x<W_xmin)   {
rcode_begin[3]=1;                 //Left
flag=1;
}

//end point of Line
if(y1>W_ymax){
rcode_end[0]=1;                 // Top
flag=1;
}
if(y1<W_ymin) {
rcode_end[1]=1;                 // Bottom
flag=1;
}
if(x1>W_xmax){
rcode_end[2]=1;                 // Right
flag=1;
```

```
rcode_end[3]=1;              //Left
flag=1;
 }
if(flag==0)
{
printf("No need of clipping as it is
already in window");
}
flag=1;
for(i=0;i<4;i++){
region_code[i]= rcode_begin[i] &&
rcode_end[i] ;
if(region_code[i]==1)
 flag=0;
}
if(flag==0)
{
printf("\n Line is completely outside
the window");
}
else{
```

```
the window");
}
else{
slope=(float)(y1-y)/(x1-x);
if(rcode_begin[2]==0 &&
rcode_begin[3]==1)   //left
{
y=y+(float) (W_xmin-x)*slope ;
x=W_xmin;

}
if(rcode_begin[2]==1 &&
rcode_begin[3]==0)       // right
{
y=y+(float) (W_xmax-x)*slope ;
x=W_xmax;

}
if(rcode_begin[0]==1 &&
rcode_begin[1]==0)      // top
{
```

```
{
x=x+(float) (W_ymax-y)/slope ;
y=W_ymax;

}
if(rcode_begin[0]==0 &&
rcode_begin[1]==1)     // bottom
{
x=x+(float) (W_ymin-y)/slope ;
y=W_ymin;

}
// end points
if(rcode_end[2]==0 && rcode_end[3]==1)
//left
{
y1=y1+(float) (W_xmin-x1)*slope ;
x1=W_xmin;
```

```
// right
{
y1=y1+(float) (W_xmax-x1)*slope ;
x1=W_xmax;

}
if(rcode_end[0]==1 && rcode_end[1]==0)
// top
{
x1=x1+(float) (W_ymax-y1)/slope ;
y1=W_ymax;

}
if(rcode_end[0]==0 && rcode_end[1]==1)
// bottom
{
x1=x1+(float) (W_ymin-y1)/slope ;
y1=W_ymin;

}
```

```
}
if(rcode_end[0]==0 && rcode_end[1]==1)
// bottom
{
x1=x1+(float) (W_ymin-y1)/slope ;
y1=W_ymin;

}
}
delay(1000);
clearviewport();
rectangle(W_xmin,W_ymin,W_xmax,W_ymax)
;
line(0,0,600,0);
line(0,0,0,600);
setcolor(RED);
line(x,y,x1,y1);
getch();
```

```
First enter XMax, YMax : 450 450
```